



Topic : Wildlife Safari Trip Management System

Group no : MLB_PG.10.01_08

Campus : Malabe

Submission Date : 20/10/2020

We declare that this is our own work and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT20210892	S.J. Dharmagunaratna	0772075978
IT20209070	W.M.T.R. Singh	0719726803
IT20211264	M.A.D.D.K. Harischandra	0711924696
IT20215088	N.D.G. Liyanage	0710557295
IT20211578	H.M.P. Dulanjaya Premarathne	0717051129

Exercise 1

The **Wild Life Safari Trip Management System** is an online **system** that gives an introduction to **wild life**.

Any **user** can log into this **application** efficiently from any **smart device** and save a lot of time by using this application.

The application can be used to know the information regarding **safari tours**, Such as the **cost**, **locations**. After **customers** who are registered into the system as a **registered user**, if the user needs to arrange a safari tour, he needs to re-register for a safari trip.

After doing so, he has to select the location, the **package** (**basic**, **standard**, **or premium**) and finally select the **nationality** (**Sri Lankan**/Foreign).

After user submits these **details**, they are collected by the **receptionist** and submitted to the **Finance Manager**, who then decides on a **discount** after considering the package and the nationality of the user.

Receptionists, IT Managers, Finance Managers, Drivers, Photographers are the **staff** members.

Further, if the user requires a **photographer**, he can make a **request** through the application and then a photographer can accept the request.

Before the photographer accepts the request, it is important that he verify the details of the customer.

After the photographer accepts the request, he has to send the details to the receptionist.

Then receptionist calculate the cost of the package, the **driver's cost** and the **Photographer's cost** and the approved discount, the Finance Manager then finalises the **payment**.

After the payment is finalized, it is sent to the receptionist and the user via **email**.

The payment can be done by **credit card**, **debit card** or through **PayPal**. After the payment is completed the user sends the **payment proof** to the receptionist via email.

Then receptionist has to log into the official email. After that receptionist has to check all the new emails, and download the payment proof.

Then receptionist has to log into the **wallet page**. Also checks the payment method and then verify that method.

After the receipt of payment is confirmed by verifying against the payment proof, the **ticket** and the **map** are issued to the user.

When the driver is taking the customer, he needs to feed into the system the location from time to time. Similarly, he should also inform the system of any **challenges** he might face.

In case of any emergency, the **driver** must contact the receptionist immediately and then the receptionist sends an emergency team to the exact location to assist **the party**.

Then photographer takes the **photos** and at the end of the trip, they can obtain the photos and **receipt** from the system and log-out.

If he likes, the user can also give a **feedback** based on his **experience**.

If the user wishes so, he can either get an **album** or individual photos depending on his requirement by making a request to the photographer.

The **IT manager** is the one who maintains the system frequently. The sample photos of the safari, photos and **video clips** are all uploaded to the system by the IT manager.

At the end of the **month**, the **monthly report** is made by the Finance Manager.

Noun/Verb Analysis

Wild Life Safari Trip Management System	Outside scope of system
System	Outside scope of system
Wild life	Outside scope of system
User	Class
Application	Redundant
Smart device	Outside scope of system
Safari tours	Class
Cost	An attribute
Location	An attribute
Customers	Meta language
Registered user	Class
Package	Class
Basic	An attribute
Standard	An attribute
Premium	An attribute
Nationality	An attribute
Details	An attribute
Staff	Class
Receptionist	Class
Finance Manager	Class
Discount	Class
Photographer	Class
Request	An event or operation
Driver's cost	An attribute
Photographer's cost	An attribute
Payment	Class
Credit card	Class
Debit card	Class
PayPal	Class
Payment proof	Class
Wallet page	Outside scope of system
Ticket	Class
Map	An attribute
Challenges	Outside scope of system
Driver	Class
The party	Outside scope of system

Photos	An attribute
Receipt	Class
Feedback	Class
Experience	Outside scope of system
Album	Class
IT manager	Class
Video clips	An attribute
Month	Outside scope of system
Monthly report	Class

Exercise 2

User	
Responsibilities	Collaboration
Log into the System	

Safari tours	
Responsibilities	Collaboration

Registered user	
Responsibilities	Collaboration
Re-register for a safari trip	
Select the location	
Select the package	Package
Select the nationality	
Requires a photographer	Photographer
User sends the payment proof	Payment Proof, Receptionist
Give a feedback	Feedback
Get an album or individual photos	Album, Photographer

Package	
Responsibilities	Collaboration

Receptionist	
Responsibilities	Collaboration
Collect details	Registered user
Submit details	IT Manager
Calculate the cost of the package	Package
Log into the official email	
Check all the new emails	
Download the payment proof	
Log into the wallet page	
Checks the payment method	Payment proof
Verify these payment details	
Issue the map and tickets	

Staff	
Responsibilities	Collaboration

Finance Manager	
Responsibilities	Collaboration
Collect the details	Receptionist
Decides the discount	Payment, Package
Finalizes the payment	Payment
Create monthly report	Monthly Report

Discount	
Responsibilities	Collaboration

Photographer	
Responsibilities	Collaboration
Accept the request	Registered User
Verify the details of the customer	Registered User
Send the details	Receptionist
Send the details to the receptionist and user	Receptionist, User
Take the photos	

Payment	
Responsibilities	Collaboration

Credit card	
Responsibilities	Collaboration

Debit card	
Responsibilities	Collaboration

Pay Pal	
Responsibilities	Collaboration

Payment proof	
Responsibilities	Collaboration

Ticket	
Responsibilities	Collaboration

Driver	
Responsibilities	Collaboration
Contact the receptionist	Receptionist
Send the location	Receptionist

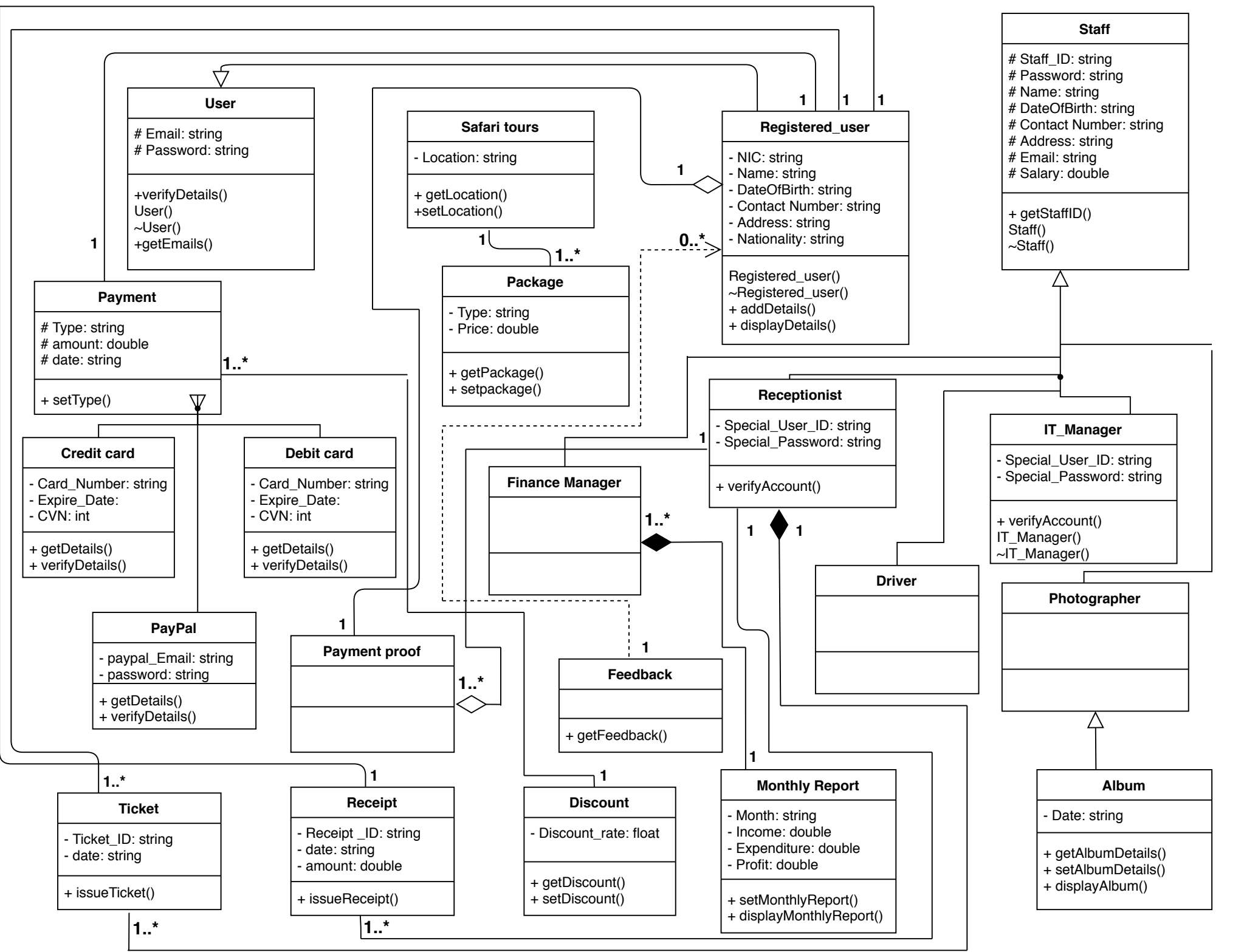
Receipt	
Responsibilities	Collaboration

Feedback	
Responsibilities	Collaboration

Album	
Responsibilities	Collaboration

IT Manager	
Responsibilities	Collaboration
Maintain the system	
Upload sample photos and videos to the system	

Monthly Report	
Responsibilities	Collaboration



Exercise 4

Main.cpp

```
#include <iostream>
#include <string>

#include "Staff.h"
#include "Finance_Manager.h"
#include "Driver.h"
#include "Photographer.h"
#include "Receptionist.h"
#include "IT_Manager.h"
#include "User.h"
#include "Registered_User.h"
#include "Ticket.h"
#include "Safari_Trip.h"
#include "Receipt.h"
#include "Package.h"
#include "Monthly_Report.h"
#include "Discount.h"
#include "Album.h"
#include "payment.h"
#include "Debit_card.h"
#include "paypal.h"
#include "Credit_Crard.h"
#include "Feedback.h"

using namespace std;

int main()
{
    Staff* member1 = new Finance_Manager("STAFF0045");
    member1->setDetails("STAFF0045", "Poorna", "22179@123", "1990-10-15", "071-8645312",
"poorna123@gmail.com", 55000.00);
    member1->displayDetails();

    Staff* member2 = new Driver("STAFF0103");
    member2->setDetails("STAFF0103", "Amarasinha", "amare45#", "1975-08-10", "072-
3876903", "amare15416@gmail.com", 33000.00);
    member2->displayDetails();

    Staff* member3 = new Photographer("STAFF0532");
    member3->setDetails("STAFF0103", "Mihindu", "56$mihi@2000", "2000-10-05", "078-
4563245", "mihphotography@outlook.com", 43000.00);
    member3->displayDetails();

    Staff* member4 = new Receptionist("STAFF0834");
    member4->setDetails("STAFF0103", "Oshani", "oshi33%$2", "1998-02-22", "077-2564321",
"oshiTharuka98@gmail.com", 40000.00);
    member4->displayDetails();

    /*Receptionist member4other;
    member4other.setReceptionistDetails("DatabaseRecep1", "dataBase123");
```

```

member4other.displayReceptionistDetails();*/

Staff* member5 = new IT_Manager("STAFF08764");
member5->setDetails("STAFF08764", "Dulaksha", "Dulak35%$3", "2098-02-12", "077-
25353321", "dulaksha12398@gmail.com", 50000.00);
member5->displayDetails();

// User* member6 = new Registered_User("abc@gmail.com");
//member6->setDetails("abc@gmail.com", "2000510", "Ajith", "2000.10.11", "0772075986",
"Gampaha", "Local");
//member6->displayDetails();

//Ticket tobject("456856");
//Receptionist robject("STAFF0103", "Oshani", "oshi33%$2", "1998-02-22", "077-
2564321", "oshiTharuka98@gmail.com", 40000.00);

Ticket t1;

string s1;
string s2;

cout << "Enter Ticket ID : " << endl;
cin >> s1;

cout << "Enter Issued Date : " << endl;
cin >> s2;

t1.issueTicket(s1, s2);

Safari_Trip ss1;

string ss2;

cout << "Enter Your Location : " << endl;
cin >> ss2;

ss1.setlocation(ss2);
ss1.getlocation();

Staff* member10 = new Finance_Manager("STAFF0047");
member10->setDetails("STAFF0047", "Kaushika", "th$456op", "1995-12-15", "070-
9645642", "Kdisanayaka16@yahoo.com", 53000.00);
member10->displayDetails();

Receipt r1;

string sid;
string sdate;
double samount;

cout << "Enter receipt ID : " << endl;
cin >> sid;

cout << "Enter date : " << endl;
cin >> sdate;

cout << "Enter amount : " << endl;
cin >> samount;

```

```

r1.issueReceipt(sid, sdate, samount);

Package p1;

string tt1;

cout << "Enter the package type (Premium/ Standard or Basic) : " << endl;
cin >> tt1;

p1.setPackage(tt1);
p1.getPackage();

Monthly_Report m1;

string mm1;
double mm2;
double mm3;
double mm4{};

cout << "Enter Month : " << endl;
cin >> mm1;

cout << "Enter Income : " << endl;
cin >> mm2;

cout << "Enter Expenditure : " << endl;
cin >> mm3;

m1.setMonthlyReport(mm1, mm2, mm3, mm4);
m1.DisplayMonthlyReport();

Discount d1;
float f1;

cout << "Enter the discount rate : " << endl;
cin >> f1;

d1.setDiscount(f1);
d1.getDiscount();

//-----//

payment pp1;
string pp2;
string pp3;
double pp4;

cout << "Enter your payment method (Credit/ Debit or PayPal) : " << endl;
cin >> pp2;
cout << "Enter your date : " << endl;
cin >> pp3;
cout << "Enter your ammount : " << endl;

```

```

cin >> pp4;

pp1.setType(pp2, pp3, pp4);
pp1.displayDetails();

Credit_Crard c1;

string cc1;
string cc2;
int cc3;

cout << "Enter your credit card number : " << endl;
cin >> cc1;

cout << "Enter the expire date : " << endl;
cin >> cc2;

cout << "Enter cvv : " << endl;
cin >> cc3;

c1.verifyDetails(cc1, cc2, cc3);
c1.getDetails();


Debit_Crard w1;

string j1;
string j2;
int j3;

cout << "Enter your credit card number : " << endl;
cin >> j1;

cout << "Enter the expire date : " << endl;
cin >> j2;

cout << "Enter cvv : " << endl;
cin >> j3;

c1.verifyDetails(j1, j2, j3);
c1.getDetails();

/*    paypal r1;

    string b1;
    string b2;

    cout << "Enter your Email : " << endl;
    cin >> b1;

    cout << "Enter Your Password : " << endl;
    cin >> b2;

    r1.verifyDetails(b1, b2);
    r1.getDetails();*/

/* Feedback ff1;

```



```
string ss1;

cout << "Enter a feedback : " << endl;
cin >> ss1;

ff1.displayFeedback(ss1);*/

return 0;

}
```

User.h

```
#ifndef USER_H
#define USER_H

#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class User
{
protected:
    string Email;
    string Password;
    string nic, name, dob, contactno, address, nationality;
public:
    void verifyDetails(string e1, string e2);
    User(string Email);
    ~User();
    string getEmail();
    virtual void setDetails(string pnic, string pname, string pdob, string pcontactno,
string paddress, string pnationality);
    virtual void displayDetails();
};

#endif
```

User.cpp

```
#include "User.h"
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

void User::verifyDetails(string e1, string e2)
{
    Email = e1;
    Password = e2;

    cout << "Verified User!" << endl;
}

User::User(string Email)
{
    this->Email = "Non";
    this->Password = "Non";
    this->nic = "Non";
    this->name = "Non";
    this->dob = "Non";
    this->contactno = "Non";
    this->address = "Non";
    this->nationality = "Non";
}

User::~~User()
{
}

string User::getEmail()
{
    return Email;
}

void User::setDetails(string pnic, string pname, string pdob, string pcontactno, string
paddress, string pnationality)
{
    this->nic = pnic;
    this->name = pname;
    this->dob = pdob;
    this->contactno = pcontactno;
    this->address = paddress;
    this->nationality = pnationality;
}

void User::displayDetails()
{
    cout << "          User Account Details" << endl;
```

```
cout << "@=====@" << endl;
cout << "NIC      \t" << nic << endl;
cout << "Name      \t" << name << endl;
cout << "Date of Birth \t" << dob << endl;
cout << "Contact Number \t" << contactno << endl;
cout << "Address \t" << address << endl;
cout << "Nationality   \t" << nationality << endl;
cout << "@=====@" << endl;
}
```

Ticket.h

```
#ifndef TICKET_H
#define TICKET_H

#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class Ticket
{
private:
    string Ticket_ID;
    string Date;

public:
    //Ticket(string pTicketID);
    //~Ticket();
    //void printTicket();
    void issueTicket(string tid, string date);
};

#endif
```

Ticket.cpp

```
#include "Ticket.h"
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

void Ticket::issueTicket(string tid, string date)
{
    Ticket_ID = tid;
    Date = date;

    cout << "Ticket ID : " << Ticket_ID << endl;
    cout << "Issued Date : " << Date << endl;
}

/*
Ticket::Ticket(string pTicketID)
{
    Ticket_ID = pTicketID;
}

Ticket::~~Ticket()
{
}

void Ticket::printTicket()
{
    cout << "Ticket ID : " << Ticket_ID << endl;
}
*/
```

Staff.h

```
#ifndef STAFF_H
#define STAFF_H

#include <string>
#include <iostream>
    using namespace std;

    class Staff
    {
    protected:
        string staffID, name, password, dateOfBirth, contactNo, email;
        double salary;
    public:
        Staff(string staffID);
        ~Staff();
        string getStaffID();
        virtual void displayDetails(void);
        virtual void setDetails(string pstaffID, string pname, string ppassword,
string pdateOfBirth, string pcontactNo, string pemail, double psalary);

    };
#endif
```

Staff.cpp

```
#include "Staff.h"
#include <string>
#include <iostream>

using namespace std;

//Constructor
Staff::Staff(string staffID)
{
    this->staffID = "0";
    this->name = "Non";
    this->password = "Non";
    this->dateOfBirth = "Non";
    this->contactNo = "Non";
    this->email = "Non";
    this->salary = 0;
}

//Destructor
Staff::~~Staff()
{
    cout << "Deleted Successfully" << staffID << endl;
}

string Staff::getStaffID()
{
    return staffID;
}

void Staff::displayDetails(void)
{
    cout << "          Staff Account Details" << endl;
    cout << "@=====@" << endl;
    cout << "Staff ID      \t" << staffID << endl;
    cout << "Name          \t" << name << endl;
    cout << "Password      \t" << password << endl;
    cout << "Date of Birth \t" << dateOfBirth << endl;
    cout << "Contact Number \t" << contactNo << endl;
    cout << "Email         \t" << email << endl;
    cout << "Salary        \t" << "Rs." << salary << ".00" << endl;
    cout << "@=====@" << endl;
}

void Staff::setDetails(string pstaffID, string pname, string ppassword, string
pdateOfBirth, string pcontactNo, string pemail, double psalary)
{
    this->staffID = pstaffID;
    this->name = pname;
    this->password = ppassword;
    this->dateOfBirth = pdateOfBirth;
    this->contactNo = pcontactNo;
    this->email = pemail;
    this->salary = psalary;
}
```


Safari_Trip.h

```
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class Safari_Trip
{
private:
    string location;
public:
    void setlocation(string s1);
    void getlocation();
};
```

Safari_Trip.cpp

```
#include "Safari_Trip.h"
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

void Safari_Trip::setlocation(string s1)
{
    location = s1;
}

void Safari_Trip::getlocation()
{
    cout << "Location is : " << location << endl;
}
```

Registered_User.h

```
#ifndef REGISTERED_USER_H
#define REGISTERED_USER_H
#include <iostream>
#include <string>
//Inheritance

#include "User.h"

class Registered_User : public User
{
    Registered_User(string Email) : User(Email) {};
    ~Registered_User();
    void addDetails(string pEmail, string pnrc, string pname, string pdob, string
pcontactno, string paddress, string pnationality);
    void displayDetails();
};

#endif
```

Registered_User.cpp

```
#include "Registered_User.h"

Registered_User::~Registered_User()
{
}

void Registered_User::addDetails(string pEmail, string pnic, string pname, string pdob,
string pcontactno, string paddress, string pnationality)
{
    this->Email = pEmail;
    this->nic = pnic;
    this->name = pname;
    this->dob = pdob;
    this->contactno = pcontactno;
    this->address = paddress;
    this->nationality = pnationality;
}

void Registered_User::displayDetails()
{
    cout << "    Registered User Account Details" << endl;
    cout << "@=====@" << endl;
    cout << "NIC          \t" << nic << endl;
    cout << "Name          \t" << name << endl;
    cout << "Date of Birth \t" << dob << endl;
    cout << "Contact Number \t" << contactno << endl;
    cout << "Address \t" << address << endl;
    cout << "Nationality    \t" << nationality << endl;
    cout << "@=====@" << endl;
}
```

Receptionist.h

```
#ifndef RECEPTIONIST_H
#define RECEPTIONIST_H

#include <iostream>
#include <string>

//Inheritance
#include "Staff.h"
//Composition
#include "Ticket.h"
class Receptionist : public Staff
{
private:
    string* Special_User_ID;
    string* Special_Password;
public:
    Receptionist(string staffID) : Staff(staffID) {};
    ~Receptionist();
    void addDetails(string pstaffID, string pname, string ppassword, string
pdateOfBirth, string pcontactNo, string pemail, double psalary);
    void displayDetails();
    void setReceptionistDetails(string pSpecial_User_ID, string pSpecial_Password);
    void displayReceptionistDetails();
};

#endif
```

Receptionist.cpp

```
#include "Receptionist.h"

Receptionist::~Receptionist()
{
}

void Receptionist::addDetails(string pstaffID, string pname, string ppassword, string
pdateOfBirth, string pcontactNo, string pemail, double psalary)
{
    this->staffID = pstaffID;
    this->name = pname;
    this->password = ppassword;
    this->dateOfBirth = pdateOfBirth;
    this->contactNo = pcontactNo;
    this->email = pemail;
    this->salary = psalary;
}

void Receptionist::displayDetails()
{
    cout << "          Staff Account Details (Receptionist)" << endl;
    cout << "@=====@" << endl;
    cout << "    Staff ID      \t" << staffID << endl;
    cout << "    Name          \t" << name << endl;
    cout << "    Password      \t" << password << endl;
}
```

```

    cout << "    Date of Birth    \t" << dateOfBirth << endl;
    cout << "    Contact Number  \t" << contactNo << endl;
    cout << "    Email            \t" << email << endl;
    cout << "    Salary          \t" << "Rs." << salary << ".00" << endl;
}

void Receptionist::setReceptionistDetails(string pSpecial_User_ID, string
pSpecial_Password)
{
    *Special_User_ID = pSpecial_User_ID;
    *Special_Password = pSpecial_Password;
}

void Receptionist::displayReceptionistDetails()
{
    cout << "    Special User ID    \t" << Special_User_ID << endl;
    cout << "    Special Password  \t" << Special_Password << endl;
    cout << "@=====@" << endl;
}

```

Receipt.h

```
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class Receipt
{
private:
    string Receipt_ID;
    string date;
    double amount;
public:
    void issueReceipt(string receipt, string d, double am);
};
```

Receipt.cpp

```
#include "Receipt.h"

void Receipt::issueReceipt(string receipt, string d, double am)
{
    Receipt_ID = receipt;
    date = d;
    amount = am;

    cout << "Receipt ID : " << Receipt_ID << endl;
    cout << "Date : " << date << endl;
    cout << "Amount : " << amount << endl;
}
```

Photographer.h

```
#ifndef PHOTOGRAPHER_H
#define PHOTOGRAPHER_H

#include <iostream>
#include <string>

//Inheritance
#include "Staff.h"
class Photographer : public Staff
{
public:
    Photographer(string staffID) : Staff(staffID) {};
    ~Photographer();
    void addDetails(string pstaffID, string pname, string ppassword, string
pdateOfBirth, string pcontactNo, string pemail, double psalary);
    void displayDetails();
};

#endif
```

Photographer.cpp

```
#include "Photographer.h"

Photographer::~Photographer()
{
}

void Photographer::addDetails(string pstaffID, string pname, string ppassword, string
pdateOfBirth, string pcontactNo, string pemail, double psalary)
{
    this->staffID = pstaffID;
    this->name = pname;
    this->password = ppassword;
    this->dateOfBirth = pdateOfBirth;
    this->contactNo = pcontactNo;
    this->email = pemail;
    this->salary = psalary;
}

void Photographer::displayDetails()
{
    cout << "          Staff Account Details (Photographer)" << endl;
    cout << "@=====@" << endl;
    cout << "    Staff ID      \t" << staffID << endl;
    cout << "    Name          \t" << name << endl;
    cout << "    Password      \t" << password << endl;
    cout << "    Date of Birth \t" << dateOfBirth << endl;
    cout << "    Contact Number \t" << contactNo << endl;
    cout << "    Email         \t" << email << endl;
    cout << "    Salary        \t" << "Rs." << salary << ".00" << endl;
    cout << "@=====@" << endl;
}
```

Paypal.h

```
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class paypal
{
private:
    string paypal_Email;
    string password;
public:
    void verifyDetails(string c1, string c2);
    void getDetails();
};
```

Paypal.cpp

```
#include "paypal.h"
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

void paypal::verifyDetails(string c1, string c2)
{
    paypal_Email = c1;
    password = c2;
}

void paypal::getDetails()
{
    cout << "PayPal Email : " << paypal_Email << endl;
    cout << "Password : " << password << endl;
}
```


Payment_Proof.h

```
class Payment_Proof
{
};
```

Payment_Proof.cpp

```
#include "Payment_Proof.h"
```

Payment.h

```
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class payment
{
protected:
    string type, date;
    double amount;
public:
    void setType(string p1, string p2, double p3);
    void displayDetails();
};
```

Payment.cpp

```
#include "payment.h"
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

void payment::setType(string p1, string p2, double p3)
{
    type = p1;
    date = p2;
    amount = p3;
}

void payment::displayDetails()
{
    cout << "User Payment Details" << endl;
    cout << "@=====@" << endl;
    cout << "Payment method is \t" << type << endl;
    cout << "Amount\t" << amount << endl;
    cout << "Card Date\t" << date << endl;
}
```

Package.h

```
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class Package
{
private:
    string type;
    double price;
public:
    void setPackage(string t1);
    void getPackage();
};
```

Package.cpp

```
#include "Package.h"
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

void Package::setPackage(string t1)
{
    type = t1;
}

void Package::getPackage()
{
    cout << "Selected package is : " << type << endl;

    if (type == "Premium")
    {
        cout << "Amount is 5000 " << endl;
    }

    else if (type == "Standard")
    {
        cout << "Amount is 2000" << endl;
    }

    else if (type == "Basic")
    {
        cout << "Amount is 1000" << endl;
    }
    else
        cout << "Wrong Package Type " << endl;
}
```

Monthly_Report.h

```
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class Monthly_Report
{
private:
    string Month;
    double Income;
    double Expenditure;
    double Profit;

public:
    void setMonthlyReport(string m, double i, double e, double p);
    void DisplayMonthlyReport();
};
```

Monthly_Report.cpp

```
#include "Monthly_Report.h"
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

void Monthly_Report::setMonthlyReport(string m, double i, double e, double p)
{
    Month = m;
    Income = i;
    Expenditure = e;
    Profit = p;
}

void Monthly_Report::DisplayMonthlyReport()
{
    double p1;

    cout << "Month : " << Month << endl;
    cout << "Income : " << Income << endl;
    cout << "Expenditure : " << Expenditure << endl;

    p1 = Income - Expenditure;
    cout << "Profit : " << p1 << endl;
}
```

IT_Manager.h

```
#ifndef IT_MANAGER_H
#define IT_MANAGER_H

#include <iostream>
#include <string>

//Inheritance
#include "Staff.h"
class IT_Manager : public Staff
{
private:
    string* Special_User_ID;
    string* Special_Password;
public:
    IT_Manager(string staffID) : Staff(staffID) {};
    ~IT_Manager();
    void addDetails(string pstaffID, string pname, string ppassword, string
pdateOfBirth, string pcontactNo, string pemail, double psalary);
    void displayDetails();
    void setReceptionistDetails(string pSpecial_User_ID, string pSpecial_Password);
    void displayReceptionistDetails();
};

#endif
```

IT_Manager.cpp

```
#include "IT_Manager.h"

IT_Manager::~IT_Manager()
{
}

void IT_Manager::addDetails(string pstaffID, string pname, string ppassword, string
pdateOfBirth, string pcontactNo, string pemail, double psalary)
{
    this->staffID = pstaffID;
    this->name = pname;
    this->password = ppassword;
    this->dateOfBirth = pdateOfBirth;
    this->contactNo = pcontactNo;
    this->email = pemail;
    this->salary = psalary;
}

void IT_Manager::displayDetails()
{
    cout << "          Staff Account Details (IT Manager)" << endl;
    cout << "@=====@" << endl;
    cout << "    Staff ID      \t" << staffID << endl;
    cout << "    Name          \t" << name << endl;
    cout << "    Password      \t" << password << endl;
    cout << "    Date of Birth \t" << dateOfBirth << endl;
    cout << "    Contact Number \t" << contactNo << endl;
    cout << "    Email         \t" << email << endl;
    cout << "    Salary        \t" << "Rs." << salary << ".00" << endl;
}

void IT_Manager::setReceptionistDetails(string pSpecial_User_ID, string
pSpecial_Password)
{
    *Special_User_ID = pSpecial_User_ID;
    *Special_Password = pSpecial_Password;
}

void IT_Manager::displayReceptionistDetails()
{
    cout << "    Special User ID \t" << Special_User_ID << endl;
    cout << "    Special Password \t" << Special_Password << endl;
    cout << "@=====@" << endl;
}
```

Finance_Manager.h

```
#ifndef FINANCE_MANAGER_H
#define FINANCE_MANAGER_H

#include <iostream>
#include <string>

//Inheritance
#include "Staff.h"
class Finance_Manager : public Staff
{
public:
    Finance_Manager(string staffID) : Staff(staffID) {};
    ~Finance_Manager();
    void addDetails(string pstaffID, string pname, string ppassword, string
pdateOfBirth, string pcontactNo, string pemail, double psalary);
    void displayDetails();
};

#endif
```

Finance_Manager.cpp

```
#include "Finance_Manager.h"
#include <iostream>
#include <string>

using namespace std;

Finance_Manager::~Finance_Manager()
{
}

void Finance_Manager::addDetails(string pstaffID, string pname, string ppassword, string
pdateOfBirth, string pcontactNo, string pemail, double psalary)
{
    this->staffID = pstaffID;
    this->name = pname;
    this->password = ppassword;
    this->dateOfBirth = pdateOfBirth;
    this->contactNo = pcontactNo;
    this->email = pemail;
    this->salary = psalary;
}

void Finance_Manager::displayDetails()
{
    cout << "          Staff Account Details (Finance Managers)" << endl;
    cout << "@=====@" << endl;
    cout << "    Staff ID      \t" << staffID << endl;
    cout << "    Name          \t" << name << endl;
    cout << "    Password      \t" << password << endl;
    cout << "    Date of Birth \t" << dateOfBirth << endl;
    cout << "    Contact Number \t" << contactNo << endl;
    cout << "    Email         \t" << email << endl;
    cout << "    Salary        \t" << "Rs." << salary << ".00" << endl;
    cout << "@=====@" << endl;
}
```


Feedback.h

```
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class Feedback
{
private:
    string feedback;

public:
    void displayFeedback(string f1);
};
```

Feedback.cpp

```
#include "Feedback.h"
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

void Feedback::displayFeedback(string f1)
{
    feedback = f1;
    cout << "Feedback : " << endl;
}
```

Driver.h

```
#ifndef DRIVER_H
#define DRIVER_H

#include <iostream>
#include <string>

//Inheritance
#include "Staff.h"
class Driver : public Staff
{
public:
    Driver(string staffID) : Staff(staffID) {};
    ~Driver();
    void addDetails(string pstaffID, string pname, string ppassword, string
pdateOfBirth, string pcontactNo, string pemail, double psalary);
    void displayDetails();
};

#endif
```

Driver.cpp

```
#include "Driver.h"

Driver::~Driver()
{
}

void Driver::addDetails(string pstaffID, string pname, string ppassword, string
pdateOfBirth, string pcontactNo, string pemail, double psalary)
{
    this->staffID = pstaffID;
    this->name = pname;
    this->password = ppassword;
    this->dateOfBirth = pdateOfBirth;
    this->contactNo = pcontactNo;
    this->email = pemail;
    this->salary = psalary;
}

void Driver::displayDetails()
{
    cout << "                Staff Account Details (Driver)" << endl;
    cout << "@=====@" << endl;
    cout << "    Staff ID      \t" << staffID << endl;
    cout << "    Name          \t" << name << endl;
    cout << "    Password      \t" << password << endl;
    cout << "    Date of Birth \t" << dateOfBirth << endl;
    cout << "    Contact Number \t" << contactNo << endl;
    cout << "    Email         \t" << email << endl;
    cout << "    Salary        \t" << "Rs." << salary << ".00" << endl;
    cout << "@=====@" << endl;
}
```

Discount.h

```
class Discount
{
private:
    float Discount_rate;
public:
    void setDiscount(float dr);
    void getDiscount();
};
```

Discount.cpp

```
//Discount.cpp file

#include "Discount.h"
#include <iostream>
#include <cstring>
using namespace std;

void Discount::setDiscount(float dr)
{
    Discount_rate = dr;
}

void Discount::getDiscount()
{
    cout << "Discount rate is : " << Discount_rate << endl;
}
```

Debit_card.h

```
#include <iostream>
#include <cstring>
#include <string>

using namespace std;
class Debit_Crard
{
private:
    string Card_Number;
    string Expire_Date;
    int Cvv;
public:
    void verifyDetails(string a1, string a2, int a3);
    void getDetails();
};
```

Debit_card.cpp

```
#include "Debit_card.h"
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

void Debit_Crard::verifyDetails(string a1, string a2, int a3)
{
    Card_Number = a1;
    Expire_Date = a2;
    Cvv = a3;
}

void Debit_Crard::getDetails()
{
    cout << "Card Number : " << Card_Number << endl;
    cout << "Expire Date : " << Expire_Date << endl;
    cout << "CVV \t" << Cvv << endl;
}
```

Credit_Crard.h

```
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class Credit_Crard
{
private:
    string Card_Number;
    string Expire_Date;
    int Cvv;
public:
    void verifyDetails(string s1, string s2, int s3);
    void getDetails();
};
```

Credit_Crard.cpp

```
#include "Credit_Crard.h"
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

void Credit_Crard::verifyDetails(string s1, string s2, int s3)
{
    Card_Number = s1;
    Expire_Date = s2;
    Cvv = s3;
}

void Credit_Crard::getDetails()
{
    cout << "Card Number : " << Card_Number << endl;
    cout << "Expire Date : " << Expire_Date << endl;
    cout << "CVN \t" << Cvv << endl;
}
```

Album.h

```
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class Album
{
private:
    string Date;
public:
    void setAlbumDetails(string d1);
    int getAlbumDetails();
    void displayAlbum();
};
```

Album.cpp

```
#include "Album.h"
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

void Album::setAlbumDetails(string d1)
{
    Date = d1;
}

int Album::getAlbumDetails()
{
    return 0;
}

void Album::displayAlbum()
{
    cout << "Date : " << Date << endl;
    cout << "200 photos and 10 videos included in the album! " << endl;
}
```

Output

```
Microsoft Visual Studio Debug Console
Staff Account Details (Finance Managers)
=====
Staff ID      STAFF0045
Name          Poorna
Password      221798123
Date of Birth  1998-10-15
Contact Number 971-8645312
Email         poorna123@gmail.com
Salary        Rs.55000.00
=====
Staff Account Details (Driver)
=====
Staff ID      STAFF0103
Name          Amarasinha
Password      amare45#
Date of Birth  1975-08-10
Contact Number 072-3876983
Email         amare15416@gmail.com
Salary        Rs.33000.00
=====
Staff Account Details (Photographer)
=====
Staff ID      STAFF0103
Name          Mihindu
Password      56$mihi@2000
Date of Birth  2000-10-05
Contact Number 078-4563245
Email         mihuphotography@outlook.com
Salary        Rs.43000.00
=====
Staff Account Details (Receptionist)
=====
Staff ID      STAFF0103
Name          Oshani
Password      oshi33$52
Date of Birth  1998-02-22
Contact Number 077-2564321
Email         oshi1iharuk@98@gmail.com
Salary        Rs.40000.00
=====
Staff Account Details (IT Manager)
=====
Staff ID      STAFF08764
Name          Dulaksha
Password      Dulak35$53
Date of Birth  2098-02-12
Contact Number 077-25353321
Email         dulaksha12398@gmail.com
Salary        Rs.50000.00
Enter Ticket ID :
2000
```

```
Microsoft Visual Studio Debug Console
Staff Account Details (IT Manager)
=====
Staff ID      STAFF08764
Name          Dulaksha
Password      Dulak35$53
Date of Birth  2098-02-12
Contact Number 077-25353321
Email         dulaksha12398@gmail.com
Salary        Rs.50000.00
Enter Ticket ID :
2000
Enter Issued Date :
200000
Ticket ID : 2000
Issued Date : 200000
Enter Your Location :
Anuradhapura
Location is : Anuradhapura
Staff Account Details (Finance Managers)
=====
Staff ID      STAFF0047
Name          Kaushika
Password      th$456op
Date of Birth  1995-12-15
Contact Number 070-9645642
Email         Kdisanayaka16@yahoo.com
Salary        Rs.53000.00
=====
Enter receipt ID :
5320
Enter date :
200.10.11
Enter amount :
50000
Receipt ID : 5320
Date : 200.10.11
Amount : 50000
Enter the package type (Premium/ Standard or Basic) :
Premium
Selected package is : Premium
Amount is 5000
Enter Month :
May
Enter Income :
40000
Enter Expenditure :
2000
Month : May
Income : 40000
Expenditure : 2000
```

```
Microsoft Visual Studio Debug Console
May
Enter Income :
40000
Enter Expenditure :
2000
Month : May
Income : 40000
Expenditure : 2000
Profit : 38000
Enter the discount rate :
10
Discount rate is : 10
Enter your payment method (Credit/ Debit or PayPal) :
Credit
Enter your date :
2000.10.11
Enter your amount :
5000
User Payment Details
@-----@
Payment method is      Credit
Amount 5000
Card Date      2000.10.11
Enter your credit card number :
20028503359
Enter the expire date :
10.11
Enter cvv :
555
Card Number : 20028503359
Expire Date : 10.11
CVN : 555
Enter your credit card number :
25598663
Enter the expire date :
20.11
Enter cvv :
555
Card Number : 25598663
Expire Date : 20.11
CVN : 555

C:\Users\Joel\Desktop\New Code\ConsoleApplication2\Debug\ConsoleApplication2.exe (process 41312) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```